

開発者向け文書

鳥取大学 工学部 吉本芳英

平成 28 年 9 月 4 日

1 コードの書き方の規約

- Fortran 90 以降の自由形式で記述すること。
- 実効性能上の大きな理由がない限り、副プログラム（サブルーチン、関数）はすべて明示的引用仕様を持っていること。このために、これらはすべて適切なモジュールのモジュール内副プログラムとし、引用側はまず対応モジュールを `use` することで、引用仕様を取得するようにすべきである。
- プログラムのインデントなどの書式は既存のものと統一されていること。
- 原則としてモジュールで共有変数を作成することを避けること。一群の変数を複数の副プログラム間で受け渡すよい方法は、構造体を作成してそれを使うことである。
- 引数の宣言において、`intent` を宣言することが望ましい。
- `#ifdef` マクロでプログラムの機能を過度に調節するのは好ましくない。できるだけ入力ファイルで調節可能とするべきである。
- 機種別にチューニングされたコードと機種依存しないコードの分離を意識してコードを書くこと。機種別コードは別ファイルに集約することが望ましい。
- コードを付け加える、変更する際にはそのファイルの先頭の著作権表示も合わせて更新すること。
- 新機能を実装した際には、その定式化を `formalism.tex` に記録するかコードに適切な文献を記録すること。
- 他のプログラムからコードを借用する際には、
 - 借用するコードのライセンスに注意する。GPL との互換性に注意すること。なお、ライブラリなどは借用せず別体としてリンクできるようにするのが良い。
 - 借用したコードはできるだけ別なファイルに分離して管理する。
 - 借用したコードは後で分離できるように配慮する。後で xTAPP 自体のライセンス条件を変更する際に必要になる。

2 コードへの機能追加

できる限り追加した機能のテスト入力とその正常な結果をパッケージ `xTAPP-test` に追加しておくべきである。

3 プログラムにでてくる各種グリッド

3.1 3次元メッシュの表現とパックされた表現

平面波で展開されるデータには3次元メッシュの上の表現と平面波基底の長さの順に1次元で並べたパックされた表現とがある。3次元メッシュにはさらに、FFT かけられる $1:2*n$ のインデックスを持つ物と、FFT にはかけられない、 $-n:n$ のインデックスを持つ物がある。

3.2 イオンの local ポテンシャルを展開するグリッド

kkg にメッシュ点のデータが入っている。メッシュ点の総数は nkf である。 kkg は対称性でつながっている G 点をまとめて表現している。対応するイオンのポテンシャルデータは既約なものだけを並べた物としている。

3.3 イオンの $Q_{i,j}(G)$ を展開するグリッド

- kqg にメッシュ点のデータが入っている。データは整数値である。メッシュ点の総数は nkc である。
- kqg は対称性を考慮していない。つまり可約である。
- qgk は kqg を逆格子 bb を使って実数値に直した物である。

3.4 計算している系の screened local ポテンシャルを展開するグリッド

- kf にメッシュ点 G のデータが入っている。メッシュ点の総数は $nkfxa$ である。
- aff に G^2 が入っている。
- kf は対称性でつながっている G 点を識別しない。
- $vbscf()$ での $trylpt$ はメッシュ上での表現である。インデックスの範囲は $-nv[xyz]:nv[xyz]$ である。

3.5 波動関数を展開するグリッド

- kg にメッシュ点 $k + G$ のデータが入っている。メッシュ点の総数は nkm である。これらは k 点ごとに値が違う。
- 対称性は考慮されていない。

3.6 パックされた表現からメッシュ上の表現への変換

```
do ig=1, nkfxa
  i1 = kf(1,ig)
  i2 = kf(2,ig)
  i3 = kf(3,ig)
  j1 = mod(nrx2+i1,nrx2)+1
  j2 = mod(nry2+i2,nry2)+1
  j3 = mod(nrz2+i3,nrz2)+1
  ...
end do
```

とすると、 $i[123]$ は $-nv[xyz]:nv[xyz]$ のメッシュ点となり、 $j[123]$ は FFT に対応する $1:2nr[xyz]$ のメッシュ点となる。FFT に対応するメッシュではもとの負のインデックスをもつ点が $2nr[xyz]$ で折り返されて入っていることに注意。

3.7 FFT 用のメッシュでの $nl[xyz]$ と $nr[xyz]$

FFT 用のメッシュの場合、メモリ上でのレイアウトを最適化すると性能が向上する。このため配列のサイズそのもの $nl[xyz]$ と実際に使う範囲 $nr[xyz]$ が分離されている。

3.8 `fftgrid.h` で定義しているマクロ

ここには、`FFT_*` というマクロが定義されている

アーキテクチャによって FFT の性能が出る複素数の配列のメモリレイアウトが、実部と虚部で隣り合う実数の配列であるケースと、実部の配列と虚部の配列の対であるケースの二種類がある。これを切り替えるためのマクロが定義されている。本来三次元であるデータを記憶列結合で一次元化して扱うものと、三次元配列として扱うものに二種類がある。また、複数の FFT データを、次元を一つ追加して単一の配列にまとめるためのマクロもある。

また、FFT を掛ける配列はキャッシュコンフリクトやバンクコンフリクトの対策上、本来の FFT のメッシュサイズよりも少し大きめにとる必要がある。この少し大きめにとるサイズの計算をするマクロもここにある。(FFT_PAD_N[XYZ])

マクロを使う例を波動関数のフーリエ変換で示すと、

```
#include "fftgrid.h"

nlx = FFT_PAD_NX(nwx2)
nly = FFT_PAD_NY(nwy2)
nlz = FFT_PAD_NZ(nwz2)
allocate(FFT_A3D_DEFSZ(wfr,nlx,nly,nlz))

call fft3_init(nwx2,nwy2,nwz2,nlx,nly,nlz,fs)

do ig=1,nkm
  k1=kg(1,ig)
```

```

      k2=kg(2,ig)
      k3=kg(3,ig)
      k1=mod(k1+nwx2,nwx2)+1
      k2=mod(k2+nwy2,nwy2)+1
      k3=mod(k3+nwz3,nwz2)+1
      FFT_A3D_RL(wfr,k1,k2,k3) = wfg(ig,1)
      FFT_A3D_IM(wfr,k1,k2,k3) = wfg(ig,2)
end do

call fft3_bw(fs,wfr,ztmp)

```

となる。

4 プログラムにでてくる構造体

xTAPP が内部で使っている構造体について

- イオンの擬ポテンシャルデータを収容する `bspot_for_atm` (`bsplpot.F90`)
- SCF iteration の制御変数を収容する `scfctrl_struct` (`scflopp.F90`)
- 各種グリッドを収容する `meshgrid_struct` (`meshgrid.F90`)
- 各種の電子状態のエネルギー値を収容する `elec_energy` (`types.f90`)
- 各種の電子状態のエネルギー値のスケール微分を収容する `elec_senergy` (`types.f90`)
- k 点 (と spin) 一個あたりの波動関数を収容する `wfn_for_k` (`types.f90`)
- 原子ごとの $Q_{ij}(\mathbf{G})$ を収容する `qg_for_atm` (`types.f90`)
- サンプル k 点の情報を収容する `irdkpt_struct` (`subr_kpoint.F90`)

4.1 構造体 `bspot_for_atm`

name: 擬ポテンシャルの名前

nl: 軌道角運動量の数

nr: 軌道ごとのレファレンスエネルギーの数

ifncpp: 擬ポテンシャルがノルム保存なら真

ifachg: 擬ポテンシャルに有効な初期電荷データがあれば真

ifcr: Ultrasoft 儀ポテンシャルのコア電荷分布データの有無のフラグ

zo: イオンの電荷

zn: 原子核の電荷

a1,a2,a3,a4: ローカルポテンシャルの長距離成分の fit データ

rc: Non local パートのカットオフの最大値 (未使用)
nsmpl: 擬ポテンシャルの Non local パートのカットオフの最大値までの log mesh 数
vgsr: ローカルポテンシャルの短距離成分の bspline fit 係数
ccg: NLCC の内核電荷の bspline fit 係数
achg: 初期電荷分布の bspline fit 係数
ddi: イオンの D_{ij}
qqr: Q_{ij} 擬ポテンシャルファイルから読んだもの
qq: Q_{ij} 再計算したもの
vpj: $\beta_{lm}(g)$ の bspline fit 係数
qj: $Q_{ij}(\mathbf{G})$ のうち動径成分の bspline fit 係数
nprj: プロジェクタの数
ip2l: プロジェクタ番号から軌道角運動量番号への変換表
ip2r: プロジェクタ番号からリファレンス番号への変換表
ip2m: プロジェクタ番号から角運動量 (ℓ, m) を通算する番号への変換表。同一の ℓ を持つ角運動量は連続に並べる。
l2ph: 与えた軌道角運動量番号をもつ角運動量を通算する番号での最初の番号への変換表。
l2pt: 与えた軌道角運動量番号をもつ角運動量を通算する番号での最後の番号への変換表。

4.2 構造体 `irdkpt_struct`

sk: k 点の位置。逆格子の基本ベクトル単位。
kp: k 点の位置。sk に nk をかけて整数にしたもの。
nsg: その k 点を動かさない対称操作の数
ngir: その k 点を動かす既約な対称操作の数
sgrg: その k 点を動かさない対称操作の番号リスト。負の番号は時間反転操作を含めることを意味する。
irrg: その k 点を動かす対称操作で既約なものの番号リスト。負の番号は時間反転操作を含めることを意味する。

ある既約な k 点と対象操作でつながる k 点を列挙したい場合、irrg を使えば良い。またある既約な k 点のブリュアンゾーン積分での重みは ngir である。したがって、rco の代わりにこれを用いるべきである。

5 プログラムにでてくる重要変数

5.1 pby(lmax**2,maxr,nsi,nw,nks,nspin)

$$\sum_{\mathbf{G}} \beta_{\tau,p}^*(\mathbf{G}) \phi_{\mathbf{k},\sigma,n}(\mathbf{G})$$

5.2 電荷 ρ とポテンシャル V

va: 単位胞の体積 Ω

- 実空間

$$\text{rho}(\mathbf{k1}, \mathbf{k2}, \mathbf{k3}) = \rho(\mathbf{r})$$

$$\text{vr}(\mathbf{k1}, \mathbf{k2}, \mathbf{k3}) = V(\mathbf{r})$$

- 逆空間

$$\text{vg}(\mathbf{k1}, \mathbf{k2}, \mathbf{k3}) = \frac{1}{\Omega} V(\mathbf{G})$$

一般に

$$A(\mathbf{G}) = \int_{\Omega} A(\mathbf{r}) e^{-i\mathbf{G} \cdot \mathbf{r}} d\mathbf{r}$$

として、 A の数値表現を

$$A(\mathbf{k1}, \mathbf{k2}, \mathbf{k3}) = A(\mathbf{r})$$

とすると、 $A(\mathbf{k1}, \mathbf{k2}, \mathbf{k3})$ の FFT 順変換は $1/\Omega A(\mathbf{G})$ になる。

このために、プログラムではイオン種 $\text{ikd} = \tau$ のローカルポテンシャルを

$$\text{vg}(\text{ig}, \text{ikd}) = \frac{1}{\Omega} V^{\tau}(G)$$

と定義している。Partial core charge である $\text{ccg}(\text{ig}, \text{ikd})$ も同様である。

5.3 FFT の定義

- 順変換 (fft3_fw)

$$\omega_1 = \frac{2\pi}{n_1}, \omega_2 = \frac{2\pi}{n_2}, \omega_3 = \frac{2\pi}{n_3}$$

$$H(l_1, l_2, l_3) = \frac{1}{n_1 n_2 n_3} \sum_{m_1=0}^{n_1-1} \sum_{m_2=0}^{n_2-1} \sum_{m_3=0}^{n_3-1} h(m_1, m_2, m_3) e^{-i\omega_1 l_1 m_1} e^{-i\omega_2 l_2 m_2} e^{-i\omega_3 l_3 m_3}$$

- 逆変換 (fft3_bw)

$$\omega_1 = \frac{2\pi}{n_1}, \omega_2 = \frac{2\pi}{n_2}, \omega_3 = \frac{2\pi}{n_3}$$

$$h(l_1, l_2, l_3) = \sum_{m_1=0}^{n_1-1} \sum_{m_2=0}^{n_2-1} \sum_{m_3=0}^{n_3-1} H(m_1, m_2, m_3) e^{i\omega_1 l_1 m_1} e^{i\omega_2 l_2 m_2} e^{i\omega_3 l_3 m_3}$$

- Fortran なのでプログラム上の添字は 1 足した物になる。

5.4 vpj(ig,ir,il,ikd)

vpj(ig,ir,il,ikd) は原子種 ikd, 軌道角運動量 $l = il$, リファレンス ir に対する、

$$4\pi \int_0^\infty dr r^2 j_l(kr) \beta_l(r)$$

である。 β_l は Ultrasoft Pseudo Potential での non-local part の射影ベクトル。ポテンシャルデータでは β_l は $r\beta_l$ で与えられていることに注意せよ。

$$\beta_{lm}(\mathbf{k}) := \int d^3r \beta_{lm}(\mathbf{r}) e^{i\mathbf{k}\cdot\mathbf{r}}; \quad -l \leq m \leq l$$

は、

$$i^l Y_{lm} 4\pi \int_0^\infty dr r^2 j_l(kr) \beta_l(r); \quad -l \leq m \leq l$$

で張られるので、これは β_l の逆フーリエ変換 (ブラベクトル) を計算するのに用いられる。

6 プログラムのループ変数

ig: 大きさ順にならべられた G 点。k + G でも使う。

is: スピン

iks: k 点とスピン (k 点だけでも使っている)

ik: k 点

il: 軌道角運動量 ($l=1$ が s に対応)

ir: レファレンスエネルギー

im: 磁気角運動量を s, p, ... の順にならべたもの

ib,jb: バンド番号

it: 原子の番号

ikd: 原子種の番号